

**In the Specification:**

Please amend the title on page 1 as follows:

~~SYNCHRONIZATION TECHNIQUES~~ DETECTING ACCESS TO A MEMORY  
LOCATION IN A MULTITHREADED ENVIRONMENT

Please amend the specification on page 1 by inserting before the "TECHNICAL FIELD" a new section as follows:

**CROSS-REFERENCE TO RELATED APPLICATION**

This application is a continuation of United States Patent Application No. 09/361,671 filed July 27, 1999, which is a continuation of United States Patent Application No. 09/192,207 filed November 13, 1998, which are hereby incorporated by reference.

Please amend Page 19, paragraph beginning at line 13:

In one embodiment, the demand evaluation techniques store in the address portion of the word of memory 701 the address of the function block 703. The function block contains the previous value that was stored in the word of memory 701, a pointer to the function that is to be invoked to calculate the data value, and the actual arguments to pass the function. When the trap 1 handler is invoked when the word of memory 701 is accessed, the trap handler uses that word of memory to locate the function block. If the access is to read the word of memory, then the trap 1 handler invokes the function passing the actual arguments. The function returns a calculated data value, which the trap 1 handler then stores in the word of memory 702. If the access is to write a value to the word of memory, then the trap 1 handler may recognize that access as an error. Alternatively, the value of the word of memory may be used as an argument that is passed to the function. Thus, the writing of a value to the word of memory would not cause an error. Rather, the writing of the value to the word of memory would result in the data value stored in the function block to be updated with the new value.

Please amend Page 21, paragraph beginning at line 4:

Figure 9 is a flow diagram of an example trap handler for demand evaluation. In step 901, the trap handler locks the trapping word of memory. In step 902, if the function associated with the trapping word is not found (e.g., the trap enable bit has been cleared), then another stream has accessed the trapping word of memory since the data blocked exception was raised for this stream. That other stream has already invoked the function and stored the calculated data value in the trapping word. Therefore, if the function is not found, then the trap handler performs a redo operation to effect the access to the trapping word of memory in step 902 and returns. If, however, the function is found, then the trap handler continues at step 903. In step 903, the trap handler unlocks the trapping word. In step 904, the trap handler invokes the function referenced by the function block. The function is preferably re-entrant and safe to be executed by multiple threads in parallel. The invoked function stores the calculated value in the trapping word, clears the trap 1 enable bit and unlocks the trapping word. In step 905, the trap handler performs the redo operation to complete the access to the trapping word and then returns.